STEAM – Unit 2 (COMPUTER PROGRAMMING)

Content Area: **Computer Programming (Grade 4)**
Course(s):
Time Period: **Ongoing**
Length: **Ongoing**
Status: **Published**

Big Idea

Learning computer science encourages critical thinking, logic, persistence, and creativity. It can help students excel at problem-solving in all subject areas, no matter what their age. At the root of all computer science is something called an algorithm. The word "algorithm" may sound like something complicated, but really it's just a list of instructions that someone can follow to achieve a result. To provide a solid base for computer science education, students should build a secure relationship with algorithms. (CODE.ORG, 2017)

Enduring Understanding

SWBAT write algorithms to control a digital character's motion, sound, color, and response to stimuli. SWBAT identify patterns in code.

Skills

- Using digital block coding applications, students write algorithms to control a digital character and it's surroundings. Students write algorithms to control the digital character's motion, sound, color, and response to stimuli (cues).
- Advanced students use a programming language such as Java or Swift to control a digital character and it's surroundings.
- Solve problems using sequential algorithms, debugging algorithms, creating loops in algorithms, and creating conditional algorithms.
- Practice problem solving and perseverance techniques.

Standards

8.2.5.E.1 Identify how computer programming impacts our everyday lives.
8.2.5.E.2 Demonstrate an understanding of how a computer takes input of data, processes and stores the data through a series of commands, and outputs information.
8.2.5.E.3 Using a simple, visual programming language, create a program using loops, events and procedures to generate specific output.
8.2.5.E.3 Using a simple, visual programming language, create a program using loops, events and procedures to generate specific output.
8.2.5.E.4 Use appropriate terms in conversation (e.g., algorithm, program, debug, loop, events, procedures, memory, storage, processing, software, coding, procedure, data).

Assessments

- Tracking of student achievement in Code.org's (or other online coding application's) lessons.
- Teacher observation

Resources/Instructional Materials

CODE.ORG CURRICULUM COURSE E. https://studio.code.org/s/coursee?section_id=1059613. Start coding with algorithms, loops, conditionals, and events and then you'll move on to functions. In the second part of this course, design and create a capstone project you can share with your friends and family.

Lesson 11: Introduction to Online Puzzles
In this progression, students will begin with an introduction (or review depending on the experience of your class) of Code.org's online workspace. Students will learn the basic functionality of the interface including the Run, Reset, and Step buttons. Dragging, deleting, and connecting Blockly blocks is also introduced in the beginning video. In the puzzles, students will practice their sequencing and debugging skills in Maze and Artist.

Lesson 12: Conditionals in Farmer
This lesson introduces students to while loops and if / else statements. While loops are loops that continue to repeat commands as long as a condition is true. While loops are used when the programmer doesn't know the exact number of times the commands need to be repeated, but the programmer does know what condition needs to be true in order for the loop to continue looping. If / Else statements offer flexibility in programming by running entire sections of code only if something is true, otherwise it runs something else.

Lesson 14: Build a Star Wars Game
In this lesson, students will practice using events to build a game that they can share online. Featuring R2-D2 and other Star Wars characters, students will be guided through events, then given space to create their own game.

Lesson 15: Functions: Songwriting
One of the most magnificent structures in the computer science world is the function. Functions (sometimes called procedures) are mini programs that you can use over and over inside of your bigger program. This lesson will help students intuitively understand why combining chunks of code into functions can be such a helpful practice.

Lesson 16: Functions in Artist
Students will be introduced to using functions on Code.org. Magnificent images will be created and modified with functions in Artist. For more complicated patterns, students will learn about nesting functions by calling one function from inside another.

Lesson 17: Functions in Bee
In the second round of practice with online functions, students will navigate complex paths, collect plenty of nectar, and make lots of honey.

Lesson 18: Functions in Harvester

Lesson 19: Determine the Concept
This lesson brings together concepts from the previous lessons and gives students a chance to think critically about how they would solve each problem, but without telling them which concept to apply. Students will

review basic algorithms, debugging, repeat loops, conditionals, while loops, and functions.

Lesson 20: Build a Play Lab Game
This lesson features Play Lab, a platform where students can create their own games and have interactions between characters and user input. Students will work with events to create keyboard controls. This set of puzzles will also loosely guide students through game development, but with freedom to add their own ideas.

CODE. ORG ANA AND ELSA This lesson features the Disney film Frozen. Students apply mathematical concepts to solve puzzles. They use knowledge of pixels, angles, acute angles, obtuse angles, sequencing, and patterns to solve problems.

GOOGLE CS FIRST CURRICULUM USING SCRATCH The activities below introduce students to computer science and the programming language Scratch. Different themes attract and engage students of varying backgrounds and interests. All materials are free and easy to use.https://csfirst.withgoogle.com/c/cs-first/en/curriculum.html

WEB SITES

- Code.org lesson plans
- Scratch lesson plans
- Tynker lesson plans

Modifications

Individual accommodations

• Additional support

• Adapting lessons to meet various learning styles

Integration of 21st Century Skills

Focus on the development of 21st Century Content Skills:

- Global awareness
- Civic literacy
- Health and wellness awareness
- Environmental literacy

Focus on the Development of Learning and Thinking Skills:

- Critical Thinking and Problem Solving Skills

- Communication Skills
- Creativity and Innovation Skills
- Collaboration Skills
- Information and Media Literacy Skills
- Contextual Learning Skills

Focus on the Development of Life Skills:

- Leadership
- Ethics
- Accountability
- Adaptability
- Personal Productivity
- Personal Responsibility
- People Skills
- Self Direction
- Social Responsibility

Interdisciplinary Connections

- Academic and Technical Rigor - Projects are designed to address key learning standards identified by the school or district.
- Authenticity - Projects use a real world context (e.g., community problems) and address issues that matter to the students.
- Applied Learning - Projects engage students in solving problems calling for competencies expected in high-performance work organizations (e.g.,teamwork, problem-solving, communication, etc.).
- Active Exploration - Projects extend beyond the classroom by connecting to community explorations.
- Adult Connections - Projects connect students with the wider community.
- Assessment Practices - Projects involve students in regular, performance-based exhibitions and assessments of their work; evaluation criteria reflect personal, school, and real-world standards of performance.